

Requested Patent: JP2000347863A

Title:

METHOD AND SYSTEM FOR COUNTING NON-SPECULATIVE EVENT INSIDE
SPECULATIVE PROCESSOR ;

Abstracted Patent: JP2000347863 ;

Publication Date: 2000-12-15 ;

Inventor(s): ALEXANDER ERIC MERIKASU ;

Applicant(s): INTERNATL BUSINESS MACH CORP ;

Application Number: JP20000135279 20000509 ;

Priority Number(s): ;

IPC Classification: G06F9/38; G06F11/34 ;

Equivalents: ;

ABSTRACT:

PROBLEM TO BE SOLVED: To enable a speculative processor to count a non-speculative event by counting the occurrence of a designated event, designating the event to be monitored, monitoring the designated event, generating the occurrence time value of the designated event and generating the number of times of occurrence of the designated event. SOLUTION: A system 200 is provided with a hierarchical memory 210 and a processor 250. Further, the hierarchical memory 210 is provided with a level 2 cache 212, a random access memory(RAM) 214 and a disk 216. Thus, a speculative processor is provided with plural counters for counting the occurrence of a designated event in a data processing system inside a performance monitor. The event to be monitored is designated. The designated event is monitored during the execution of an instruction due to the speculative processor. The number of times of occurrence of the designated event related to all the instructions to be executed by the speculative processor is generated and the number of times of occurrence related to the completely executed instruction is generated.

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2000-347863
(P2000-347863A)

(43) 公開日 平成12年12月15日 (2000. 12. 15)

(51) Int.Cl. ⁷	識別記号	F I	テマコード* (参考)
G 0 6 F 9/38	3 8 0	G 0 6 F 9/38	3 8 0 C
11/34		11/34	L

審査請求 有 請求項の数17 OL (全 14 頁)

(21) 出願番号 特願2000-135279(P2000-135279)
(22) 出願日 平成12年5月9日 (2000. 5. 9)
(31) 優先権主張番号 09/310912
(32) 優先日 平成11年5月13日 (1999. 5. 13)
(33) 優先権主張国 米国 (US)

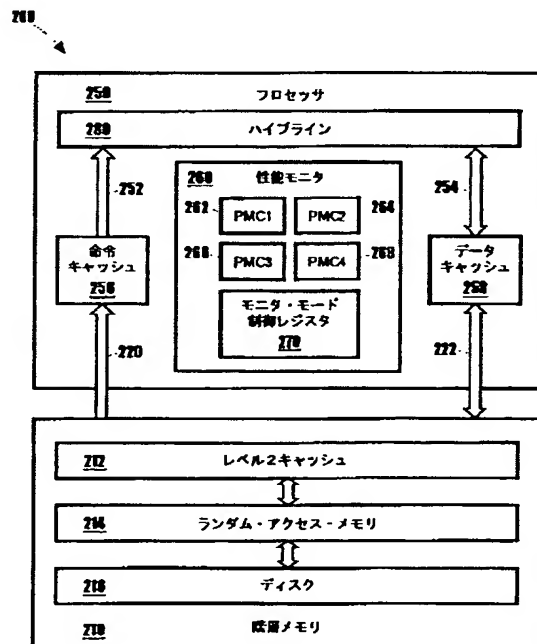
(71) 出願人 390009531
インターナショナル・ビジネス・マシーンズ・コーポレーション
INTERNATIONAL BUSINESS MACHINES CORPORATION
アメリカ合衆国10504、ニューヨーク州
アーモンク (番地なし)
(72) 発明者 アレクサンダー・エリック・メリカス
アメリカ合衆国78717 テキサス州オースチン
ラクラン・ドライブ 10037
(74) 代理人 100086243
弁理士 坂口 博 (外1名)

(54) 【発明の名称】 投機的プロセッサ内で非投機的イベントをカウントするための方法およびシステム

(57) 【要約】 (修正有)

【課題】 投機的プロセッサ内で非投機的イベントをカウントするための方法およびシステムを提供する。

【解決手段】 投機的プロセッサには、データ処理システム内の指定されたイベントの発生をカウントするために、性能モニタ内に複数のカウンタが含まれる。監視するイベントが指定されると、そのイベントは、投機的プロセッサによる命令の実行中に監視される。投機的プロセッサによって実行されるすべての命令に関する、指定されたイベントの発生回数が生成され、投機的プロセッサによって完全に実行される命令に関する、指定されたイベントの発生回数が生成される。すべての命令に関する指定されたイベントの発生回数と、すべての完了した命令に関する指定されたイベントの発生回数との間の差を、投機的プロセッサによって投機的に実行された命令に関する指定されたイベントの発生回数として生成する。



【特許請求の範囲】

【請求項1】投機的プロセッサと複数のカウンタとを含むデータ処理システム内でイベントを監視する方法であって、前記複数のカウンタの中の各カウンタが、前記データ処理システム内の指定されたイベントの発生をカウントし、

監視するイベントを指定し、

前記投機的プロセッサによる命令の実行中に前記指定されたイベントについて監視し、

前記投機的プロセッサによって実行されるすべての命令について、前記指定されたイベントの発生回数値を生成し、

前記投機的プロセッサによって完全に実行される命令について、前記指定されたイベントの発生回数を生成する方法。

【請求項2】すべての命令に関する前記指定されたイベントの発生の前記回数と、すべての完了した命令に関する前記指定されたイベントの発生の前記回数との間の差を、前記投機的プロセッサによって投機的に実行された命令の前記指定されたイベントの発生回数として計算するステップをさらに含む、請求項1に記載の方法。

【請求項3】前記投機的プロセッサによって実行される命令のそれぞれについて複数の指定されたイベントを監視するステップをさらに含む、請求項1に記載の方法。

【請求項4】投機的プロセッサと複数のカウンタとを含むデータ処理システム内でイベントを監視する方法であって、前記複数のカウンタの中の各カウンタが、前記データ処理システム内の指定されたイベントの発生をカウントし、

仮カウンタを特定の命令に関連付け、

第1グローバル・イベント・カウンタをすべての命令に関連付け、

第2グローバル・イベント・カウンタを完了した命令に関連付け、

監視するイベントを指定し、

前記投機的プロセッサによる命令の実行中に前記指定されたイベントについて監視し、

前記特定の命令の実行中の前記イベントの発生の検出に

応答して、前記仮カウンタをインクリメントし、命令のいずれかの実行中の前記イベントの発生の検出に

応答して、前記第1グローバル・イベント・カウンタをインクリメントし、前記特定の命令の完了の検出に

応答して、前記仮カウンタからのイベント回数を前記第2グローバル・イベント・カウンタに加算する方法。

【請求項5】前記第1グローバル・イベント・カウンタからのイベント回数と、前記第2グローバル・イベント・カウンタからのイベント回数との間の差を、前記投機的プロセッサによって投機的に実行された命令に関する前記指定されたイベントの発生回数として計算するステ

ップをさらに含む、請求項4に記載の方法。

【請求項6】投機的プロセッサと複数のカウンタとを含むデータ処理システム内でイベントの回数を計算する方法であって、前記複数のカウンタの中の各カウンタが、前記データ処理システム内の指定されたイベントの発生をカウントし、

前記投機的プロセッサによって実行されるすべての命令に関する指定されたイベントの発生回数を第1カウンタから読み取り、

前記投機的プロセッサによって完全に実行される命令に関する指定されたイベントの発生回数を第2カウンタから読み取り、

すべての命令に関する前記指定されたイベントの発生の前記回数と、すべての完了した命令に関する前記指定されたイベントの発生の前記回数との間の差を、前記投機的プロセッサによって投機的に実行された命令に関する前記指定されたイベントの発生回数として計算する方法。

【請求項7】投機的プロセッサと複数のカウンタとを含むデータ処理システム内でイベントを監視する方法であって、前記複数のカウンタの中の各カウンタが、前記データ処理システム内の指定されたイベントの発生をカウントし、

監視するイベントを指定し、

前記指定されたイベントを監視し、

特定の指定されたイベントの発生の検出に

応答して、第1カウンタおよび第2カウンタをインクリメントし、命令の完了の検出に

応答して、前記第2カウンタを第3カウンタに加算する方法。

【請求項8】前記第1カウンタが、前記投機的プロセッサによって実行されるすべての命令に関する特定の指定されたイベントの発生をカウントする、請求項7に記載の方法。

【請求項9】前記第2カウンタが、特定の命令に関する特定の指定されたイベントの発生をカウントする、請求項7に記載の方法。

【請求項10】前記第3カウンタが、前記投機的プロセッサによって完全に実行される命令に関する特定の指定されたイベントの発生をカウントする、請求項7に記載の方法。

【請求項11】投機的に実行された命令に関する特定の指定されたイベントの発生回数を生成するために、前記第1カウンタと前記第3カウンタとの間の差を計算するステップをさらに含む、請求項7に記載の方法。

【請求項12】投機的プロセッサと複数のカウンタとを含むデータ処理システム内でイベントを監視するための装置であって、前記複数のカウンタの中の各カウンタが、前記データ処理システム内の指定されたイベントの発生をカウントし、

監視するイベントを指定するための手段と、

前記投機的プロセッサによる命令の実行中に前記指定されたイベントについて監視するための手段と、

前記投機的プロセッサによって実行されるすべての命令に関する前記指定されたイベントの発生回数を生成するための第1手段と、

前記投機的プロセッサによって完全に実行される命令に関する前記指定されたイベントの発生回数を生成するための第2手段とを含む装置。

【請求項13】すべての命令に関する前記指定されたイベントの発生の前記回数と、すべての完了した命令に関する前記指定されたイベントの発生の前記回数との間の差を、前記投機的プロセッサによって投機的に実行された命令に関する前記指定されたイベントの発生回数として計算するための手段をさらに含む、請求項12に記載の装置。

【請求項14】前記投機的プロセッサによって実行される命令のそれぞれについて複数の指定されたイベントを監視するための手段をさらに含む、請求項12に記載の装置。

【請求項15】複数のカウンタを含む投機的プロセッサ内のイベントの回数を計算するためのシステムであって、前記複数のカウンタの中の各カウンタが、前記投機的プロセッサ内の指定されたイベントの発生をカウントし、

前記投機的プロセッサによって実行されるすべての命令に関する指定されたイベントの発生回数を第1カウンタから読み取るための第1手段と、

前記投機的プロセッサによって完全に実行される命令に関する前記指定されたイベントの発生回数を第2カウンタから読み取るための第2手段と、

すべての命令に関する前記指定されたイベントの発生の前記回数と、すべての完了した命令に関する前記指定されたイベントの発生の前記回数との間の差を、前記投機的プロセッサによって投機的に実行された命令に関する前記指定されたイベントの発生回数として計算するための手段とを含むシステム。

【請求項16】投機的プロセッサと複数のカウンタとを含むデータ処理システム内でイベントを監視するための、コンピュータ可読媒体上のコンピュータ・プログラム製品であって、前記複数のカウンタの中の各カウンタが、前記データ処理システム内の指定されたイベントの発生をカウントし、

監視するイベントを指定するための第1命令と、

前記投機的プロセッサによる命令の実行中に前記指定されたイベントについて監視するための第2命令と、

前記投機的プロセッサによって実行されるすべての命令に関する前記指定されたイベントの発生回数を生成するための第3命令と、

前記投機的プロセッサによって完全に実行される命令に関する前記指定されたイベントの発生回数を生成するた

めの第4命令とを含むコンピュータ・プログラム製品。

【請求項17】複数のカウンタを含む投機的プロセッサ内でイベントの回数を計算するための、コンピュータ可読媒体上のコンピュータ・プログラム製品であって、前記複数のカウンタの中の各カウンタが、前記投機的プロセッサ内の指定されたイベントの発生をカウントし、前記投機的プロセッサによって実行されるすべての命令に関する指定されたイベントの発生回数を第1カウンタから読み取るための第1命令と、

前記投機的プロセッサによって完全に実行される命令に関する前記指定されたイベントの発生回数を第2カウンタから読み取るための第2命令と、

すべての命令に関する前記指定されたイベントの発生の前記回数と、すべての完了した命令に関する前記指定されたイベントの発生の前記回数との間の差を、前記投機的プロセッサによって投機的に実行された命令に関する前記指定されたイベントの発生回数として計算するための第3命令とを含むコンピュータ・プログラム製品。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、全般的には改良されたデータ処理システムに関し、具体的には、データ処理システム内の性能を監視するための方法およびシステムに関する。

【0002】

【従来の技術】プロセッサを使用する通常のコンピュータ・システムでは、システム開発者は、最も効果的なシステム設計のためにソフトウェア実行の最適化を望む。通常、プログラムの、メモリに対するアクセス・パターンとシステムのメモリ階層との相互作用でのシステム効率を判定するための調査が実行される。メモリ階層の挙動の理解は、タスクのスケジューリングまたは区分を行うアルゴリズムの開発ならびに、システムを最適化するためのデータの分配および構造化に役立つ。

【0003】現況技術のプロセッサ内では、プロセッサが、ソフトウェアによって選択可能なイベントの発生をカウントし、関連するデータ処理システム内でのプロセスの実行の時間を測定することを可能にする機構が設けられることがしばしばである。これらの機構を、プロセッサの性能モニタと称する。性能監視は、システムでのソフトウェアの使用を最適化するために使用されることがしばしばである。性能モニタは、一般に、特定の時点での計算機の状態を判定することによってシステムのデバッキングおよび分析を援助するために選択された特性を監視するための、プロセッサに組み込まれた機構とみなされる。性能モニタは、プロセッサの命令実行および記憶域制御の利用に関する情報を作ることがしばしばである。たとえば、性能モニタを使用して、処理システム内のイベントの間に経過した時間の長さに関する情報を提供することができる。もう1つの例として、ソフトウ

ェア・エンジニアは、分岐命令およびメモリ・アクセスを再配置することによってプログラムを最適化するために、性能モニタからのタイミング・データを使用することができる。さらに、性能モニタは、データ処理システムのL1キャッシュ、L2キャッシュおよび主記憶へのアクセス時間に関するデータを収集するのに使用することができる。このデータを使用して、システム設計者は、特定のソフトウェア環境またはハードウェア環境に固有の性能ボトルネックを識別することができる。作られる情報は、通常は、システム設計者にとって、所与のシステムの性能強化の手段または新規システムの設計での改良の開発の手段に向かう案内になる。

【0004】データ処理システム内のイベントは、性能モニタ内の1つまたは複数のカウンタによってカウントされる。そのようなカウンタの動作は、複数のビット・フィールドからなる制御レジスタによって管理される。一般に、制御レジスタとカウンタの両方が、ソフトウェアによって読書可能である。したがって、制御レジスタに値を書き込むことによって、ユーザは、データ処理システム内の監視されるイベントを選択し、カウンタを使用可能にする条件を指定することができる。

【0005】プロセッサの効率を評価するためには、実行された作業の量と、命令実行のために消費された資源の量を判定する必要がある。多くの現代のプロセッサでは、高性能を達成するために投機的 (speculative) 処理が実施されている。その結果、処理される命令の一部は、それが投機的に実行された条件が発生しなかったので、完全に実行されずに取り消されるかフラッシュされる場合がある。他の命令と同様に、投機的命令は、プロセッサ内の資源を消費する。ほとんどの現代のプロセッサでは、資源の使用に関連する、事前に定義されたイベントの発生をカウントする性能モニタ・カウンタが実施されている。しかし、性能監視と命令の投機的実行の両方を備えるプロセッサでは、性能モニタ・カウンタが、非投機的命令すなわち実行を完了する命令と、投機的命令すなわち完了しない命令の両方についてイベントをカウントする。

【0006】

【発明が解決しようとする課題】したがって、命令の投機的実行を実行するプロセッサ内で、資源の使用を正確に監視するための方法およびシステムを有することが有利である。完了した命令と、投機的に実行され、完了しなかった命令との間で、プロセッサ内の資源の使用を比較するための方法およびシステムを有することは、さらに有利である。

【0007】

【課題を解決するための手段】本発明は、投機的プロセッサで非投機的イベントをカウントするための方法およびシステムを提供する。投機的プロセッサには、性能モニタ内に、データ処理システム内の指定されたイベント

の発生をカウントするための複数のカウンタが含まれる。監視されるイベントが指定される。指定されたイベントは、投機的プロセッサによる命令の実行中に監視される。投機的プロセッサによって実行されるすべての命令に関する、指定されたイベントの発生回数が生成され、投機的プロセッサによって完全に実行された命令に関する、指定されたイベントの発生回数が生成される。すべての命令に関する指定されたイベントの発生回数と、すべての完了した命令に関する指定されたイベントの回数の間の差を、投機的プロセッサによって投機的に実行された命令に関する指定されたイベントの発生回数として生成することができる。

【0008】

【発明の実施の形態】図1を参照すると、絵図に、本発明の好ましい実施例による、本発明を実施することのできるデータ処理システムが示されている。システム・ユニット110と、ビデオ表示端末102と、キーボード104と、フロッピー駆動装置および他の種類の永続的記憶媒体および取外し可能記憶媒体を含めることのできる記憶装置108と、マウス106とを含むパーソナル・コンピュータ100が図示されている。追加の入力装置を、パーソナル・コンピュータ100に含めることができる。パーソナル・コンピュータ100は、適当なコンピュータを使用して実施することができる。図示の表現ではパーソナル・コンピュータが示されているが、本発明の他の実施形態は、メインフレーム、ワークステーション、ネットワーク・コンピュータ、インターネット機器、パームトップ・コンピュータなどの他の種類のデータ処理システムで実施することができる。システム・ユニット110には、メモリ、中央処理装置、入出力装置などが含まれる。しかし、本発明では、システム・ユニット110に、中央処理装置またはシステム・ユニット内に存在する複数のCPUのうちの1つのいずれかとして、投機的プロセッサが含まれる。

【0009】図2を参照すると、ブロック図に、本発明の好ましい実施例による、情報を処理するためのデータ処理システムの選択された内部機能ユニットが示されている。システム200には、階層メモリ210とプロセッサ250が含まれる。階層メモリ210には、レベル2キャッシュ212、ランダム・アクセス・メモリ (RAM) 214およびディスク216が含まれる。レベル2キャッシュ212は、当技術分野で周知の形で、RAM214に格納できるデータおよび命令への高速アクセス・キャッシュを提供する。RAM214は、不揮発性のディスク216に格納されたデータおよび命令のキャッシュを提供することもできる、データおよび命令のための主記憶域を提供する。

【0010】データと命令は、階層メモリ210からプロセッサ250へ、命令転送経路220およびデータ転送経路222上で転送することができる。命令転送経路

220およびデータ転送経路222は、単一のバスまたは別々のバスとして、プロセッサ250と階層メモリ210の間で実施することができる。その代わりに、単一のバスでプロセッサ250と階層メモリ210の間でデータと命令を転送し、プロセッサ250に、命令バス252とデータ・バス254などのプロセッサ250内の別々の命令転送経路およびデータ転送経路を設けることができる。

【0011】プロセッサ250には、命令キャッシュ256、データ・キャッシュ258、性能モニタ260および命令パイプライン280も含まれる。性能モニタ260には、性能モニタ・カウンタ(PMC1)262、性能モニタ・カウンタ(PMC2)264、性能モニタ・カウンタ(PMC3)266、性能モニタ・カウンタ(PMC4)268およびモニタ・モード制御レジスタ(MMCR)270が含まれる。その代わりに、プロセッサ250は、図示されていない他のカウンタおよび制御レジスタを有することができる。

【0012】プロセッサ250には、縮小命令セット・コンピューティング(RISC)プロセッサのPowerPCファミリなど、単一サイクル内に複数の命令を実行できるパイプライン式プロセッサが含まれる。システム200の動作中に、命令およびデータは、階層メモリ210に格納される。実行される命令は、命令キャッシュ256を介して命令パイプライン280に転送される。命令キャッシュ256は、パイプライン内でステージングされた命令を復号し、実行する。一部の命令は、データ・キャッシュ258を介して階層メモリ210との間でデータを転送する。それ以外の命令は、メモリからロードされたデータを操作するか、命令の流れを制御することができる。

【0013】性能モニタ260には、PMC1 262ないしPMC4 268とMMCR270を含む、イベント検出制御論理が含まれる。性能モニタ260は、プロセッサ命令実行および記憶域制御の利用に関するかなりの粒度を有する詳細情報を提供する目的の、ソフトウェアからアクセス可能な機構である。性能モニタには、プロセッサ/記憶域関連のイベントをカウントするのに使用される、実施形態に依存する個数の性能モニタ・カウンタ(PMC)を含めることができる。これらのカウンタを、「グローバル・カウンタ」と称する場合もある。MMCRは、カウンタの機能確立し、各MMCRは通常は、ある個数のカウンタを制御する。PMCおよびMMCRは、通常は、物理的にプロセッサ上に存在する専用レジスタである。これらのレジスタは、読取動作または書込動作のために、その目的のための特殊命令を介してアクセス可能である。書込動作は、特権状態またはスーパーバイザ状態でのみ許可されることが好ましく、読取は、専用レジスタの内容を変更しないので、問題状態で許容されることが好ましい。異なる実施形態で

は、これらのレジスタは、入出力空間のアドレスなど、他の手段によってアクセス可能とすることができる。好ましい実施例では、PMC1ないしPMC4は、32ビット・カウンタであり、MMCRは、32ビット・レジスタである。カウンタおよび制御レジスタのサイズが、製造のコスト、プロセッサ250の所望の機能性およびプロセッサ250内で使用可能なチップ面積を含む設計検討事項に依存することを、当業者は諒解するであろう。

【0014】性能モニタ260は、システム全体を監視し、命令処理の結果として発生するイベントの回数を累算する。本発明では、プロセッサ250は、命令がプログラムによってコーディングされた順序またはコンパイラによってプログラムのコンパイル中に並べられた順序に関してアウトオブオーダーに命令を実行することができる。プロセッサ250は、投機的実行も使用して、ある命令が依存するデータが使用可能になる前に、その命令の条件分岐の結果を予測する。MMCRは、記録/カウントするイベント/信号の選択を考慮したビット・フィールドに区分される。イベントの許容可能な組合せを選択することによって、カウンタが並列に動作するようになる。性能モニタが、本発明によって提供される形で、投機的に実行される命令と共に使用される時には、性能モニタは、完了した命令と投機的に実行されるまだ未完了の命令の両方の実行中にプロセッサの性能を監視するための機構として使用することができる。

【0015】図3を参照すると、2つのPMCの動作を制御するのに適したMMCRの1つの構成の例を表す図が示されている。この例からわかるように、MMCRは、複数のビット・フィールドに区分され、ビット・フィールドの設定によって、カウントするイベントが選択され、性能モニタ割込みが使用可能にされ、回数を使用可能にする条件が指定される。その代わりに、MMCRに、この図に示されていない初期設定回数値をセットすることができる。

【0016】初期設定回数値は、可変かつソフトウェア選択可能である。初期設定回数値は、命令が実行のために初めてスケジューリングされる時にカウンタにロードすることができる。たとえば、調査中のイベントが「レジスタ・アクセス」であると仮定すると、初期設定回数値によって、関連する命令に関するレジスタ・アクセスの数が指示される場合に、その命令の完了によって、その特定の命令に関するレジスタ・アクセスの数を、すべての命令によるすべてのレジスタ・アクセスをカウントするPMCの総イベント回数に加算することができる。もちろん、実行されるデータ命令に応じて、「完了」が異なる意味を有する場合がある。たとえば、「load」命令の場合、「完了」は、その命令に関連するデータが受け取られたことを示すが、「store」命令の場合、「完了」は、データが成功裡に書き込まれたこと

を示す。たとえばPMC1などのユーザ読取可能カウンタが、PMC1が最初に初期設定された後のレジスタ・アクセスの総数へのソフトウェア・アクセスを提供する。適当な値を用いると、性能モニタは、簡単にシステム性能特性の識別に使用するのに適するようになる。

【0017】図3のMMCRのビット0ないし4および18によって、回数が使用可能にされるシナリオが決定される。たとえば、ビット0は、このビットがセットされている時に、PMCの値がハードウェア・イベントによって変更されない、すなわち、回数が凍結される、フリーズ回数ビットとすることができる。ビット1ないし4によって、回数が実行される他の特定の条件を示すことができる。ビット5、16および17は、PMCによってトリガされる割込み信号の制御に使用される。ビット6ないし9は、時間ベースまたはイベントベースの遷移を制御するのに使用することができる。ビット19ないし25は、PMC1のイベント選択すなわち、PMC1に関してカウントされる信号の選択に使用することができる。ビットの機能と数は、特定の実施形態で必要なイベントを選択する必要に応じて選択することができる。

【0018】なんらかの種類の性能分析用のデータを取り込むためには、少なくとも1つのカウンタが必要である。カウンタの数が増えれば、より高速またはより正確な分析がもたらされる。監視されるシナリオが、厳密に反復可能である場合には、異なる項目を選択した状態で同一のシナリオを実行することができる。シナリオが厳密に反復可能ではない場合には、同一の項目を選択した状態で同一のシナリオを複数回実行して、統計データを収集することができる。シナリオの開始からの時間は、システム時間サービスを介して使用可能であり、その結果、時間の間隔を使用して、異なるサンプルと異なるイベントを相関させることが可能になると仮定する。

【0019】図4を参照すると、アウトオブオーダー投機的実行プロセッサ内の命令パイプラインのステージの詳細を示すブロック図が示されている。システム400には、メモリ・システム402、データ・キャッシュ404、命令キャッシュ406および性能モニタ410が示されており、これらは、図3に示された階層メモリ、データ・キャッシュ、命令キャッシュおよび性能モニタに類似するものとして行うことができる。命令は、実行される際に、キャッシュ・アクセスなどのプロセッサ内のイベントを引き起こす。性能モニタ410には、1つまたは複数のMMCRの制御下でイベントをカウントする複数のPMCが含まれる。これらのカウンタとMMCRは、内部プロセッサ・レジスタであり、ソフトウェア制御の下で読み取り、書き込むことができる。

【0020】取出ユニット420は、命令キャッシュ406から命令を取り出し、命令キャッシュ406は、メモリ・システム402から命令を取り出す。復号ユニッ

ト440は、命令を復号して、命令の種類、ソース・レジスタおよびデスティネーション・レジスタなどの、命令に関する基本情報を判定する。シーケンシング・ユニット450は、復号された情報を使用して、実行のために命令をスケジューリングする。命令を追跡するために、完了テーブル460を使用して、スケジューリングされた命令に関する情報を格納し、検索する。シーケンシング・ユニット450は、ディスパッチされた命令を完了テーブル460内の関連項目に割り当てる際に、先入れ先出しベースまたは回転式で、項目を実行中の命令に割り当てるか関連付ける。命令が実行される際に、実行中の命令に関する情報が、特定の命令に関する完了テーブル460の関連項目のさまざまなフィールドおよびサブフィールドに格納される。

【0021】実行ユニット#1 482または実行ユニット#N 484などの実行ユニット1ないしNのうちの1つを使用して実行制御ユニット480によって実行される命令は、ロード/ストア・ユニット486を使用して、データ・キャッシュ404を介してメモリ・システム402にデータを書き込むか、メモリ・システム402からデータを読み取ることができる。命令が完了する際に、完了ユニット470が、命令の実行の結果をコミットし、命令のデスティネーション・レジスタが、後続命令による使用のために使用可能にされる。どの命令でも、ソース・レジスタが使用可能になるや否や、適当な実行ユニットに発行することができる。

【0022】命令は、制御（分岐）命令によって、条件的または無条件のいずれかで命令の流れが変更されるまで、順次取り出され、完了される。制御命令では、命令の取出を開始する新しいデータ位置が指定される。取出ユニット420が、条件に基づくデータがまだ使用可能でない（たとえば、必要なデータを作る命令がまだ実行されていない）条件分岐動作およびデータを受け取る時に、取出ユニット420は、分岐予測制御ユニット430内の1つまたは複数の分岐予測機構を使用して、条件の結果を予測することができる。その後、制御は、条件の結果を決定できるまで、投機的に変更される。分岐が正しく予測された場合、動作が継続する。予測が正しくなかった場合には、投機的経路に沿った命令のすべてが、取り消されるかフラッシュされる。

【0023】投機的命令は、分岐条件が解決されるまで完了できないので、多数の高性能アウトオブオーダー・プロセッサは、物理レジスタを仮想レジスタにマッピングする機構を備えている。実行の結果は、命令が実行を終了した時に、仮想レジスタに書き込まれる。物理レジスタは、命令が実際に完了するまで更新されない。前の命令の結果に依存する命令は、仮想レジスタが書き込まれるや否や、実行を開始することができる。この形で、条件分岐の結果を判定する前に、投機的命令の長いストリームを実行することができる。

【0024】図5を参照すると、投機的プロセッサ内で使用することのできる命令完了テーブルの図が示されている。完了テーブル500は、環状リストとして編成され、完了テーブル500の各項目が、単一の命令を追跡するようになっている。命令は、完了テーブル500でのインデックス値と等しい「タグ値」を有するといわれる。命令に関連するタグ値は、その命令の実行によって引き起こされるイベントの識別に使用することができる。割振りタグ502は、次の使用可能な項目のインデックスを保持する。完了タグ504は、完了する命令のインデックスを保持する。完了テーブルに使用可能な項目がない場合には、項目が使用可能になるまで、プロセッサのシーケンシング・ユニットがストールする。

【0025】図5には、完了タグが項目3を指し、割振りタグが項目6を指している、完了テーブル500内のデータの例が示されている。項目3の命令が、次に完了する。項目4および5の命令は、実行を待っているか、現在実行中であるか、完了を待っている可能性がある。次に復号される命令は、項目6に配置され、割振りポインタは、項目7を指すようにインクリメントされる。割振りポインタが項目15を指している場合には、存在しない項目16にインクリメントするのではなく、0にラップする、すなわち、ポインタは、環状の形でラップする。この例では、割振りポインタが項目2を指す場合には、使用可能な項目がなくなる。

【0026】その代わりに、単一の完了テーブル項目を、命令のグループに割り振ることができる。その場合、グループ内のすべての命令が、単一のタグ値を用いて追跡される。

【0027】図6を参照すると、完了した命令に関するイベントから分離して、投機的に実行される命令に関するイベントをカウントするためのプロセッサの図が示されている。命令の完了は、完了テーブル650を介して追跡される。各テーブル項目は、関連するタグ値652を有する。命令のタグ値は、テーブル内のその命令に関する項目の位置であり、これによって、プロセッサが、プロセッサ内の1つまたは複数のイベントを引き起こした命令を識別できるようになる。命令は、「システムを介して移動する」（または実行される）際に、イベント信号612を生成し、このイベント信号612は、特定のイベントまたは条件が発生したことを知らせる。イベント信号と共に、イベント・タグ614を使用して、どの命令がイベント信号に関連するイベントを引き起こしたかを識別する。

【0028】完了テーブル650は、項目ごとに1つまたは複数の仮カウンタ654を含むように拡張される。仮カウンタは、絶対に完了しない可能性がある（投機的実行）命令によって引き起こされる監視されるイベントの発生回数を保持する。仮カウンタの監視されるイベントは、MMCR670のフィールド672によって制御

される。フィールド672の内容は、信号613を生成して、イベント信号612をフィルタリングするマルチプレクサ610を制御するのに使用される。イベント・タグ614は、選択されたイベント信号615を、デコーダ616を介して適当な命令の仮カウンタに経路指定するのに使用される。仮カウンタ654は、デコーダ616からインクリメンタ（incrementer）618に制御信号617を送り、入力620を介して仮カウンタを読み取り、カウンタから読み取った値をインクリメントし、出力619を介してその値を仮カウンタに書き戻すことによってインクリメントされる。仮カウンタの値は、インクリメンタ618によってインクリメントされ、インクリメンタ618は、図6に示されているように、特定の仮カウンタ専用とすることができる。その代わりに、単一のインクリメンタをすべてのカウンタによって共用することができる。

【0029】1つまたは複数のPMCを使用して、1つまたは複数のMMCRのフィールドによる制御に従って監視されるイベントの発生をカウントする。フィールド674を使用して、制御信号661を生成し、その結果、PMC1664が、仮カウンタからの仮回数を受け入れるようにする。仮カウンタに格納されたイベントの仮回数は、加算器662によってPMC1664の元の値に加算され、その後、PMC1664に格納される。命令が完了した時に、完了タグ632に、完了テーブル650での完了する命令のインデックスが保持される。完了タグ632は、制御信号631によってマルチプレクサ630を制御して、MMCRのフィールド674によって制御されるマルチプレクサ660に完了する命令の仮カウンタの内容を送るのに使用される。マルチプレクサ660が、制御信号661によって適当に制御される場合に、イベント668は、仮カウンタ値を加算器662に送らせ、加算器662はPMCを更新する。

【0030】図6では、仮カウンタが、命令完了テーブルの物理的拡張として図示されている。その代わりに、仮カウンタを、完了テーブルから物理的に分離し、なおかつイベント・タグ、割振りタグおよび完了タグによってインデクシングできるようにすることができる。図6では、仮カウンタに関連するインクリメンタも図示されている。その代わりに、回数値を仮カウンタに直接に書き込むことができる。たとえば、プロセッサのロード／ストア・ユニットが、実行中の命令によって読み書きされたバイト数を書き込むことができる。単一の完了テーブル項目を使用して複数の命令のグループを追跡するプロセッサの場合、仮カウンタに、グループ内の命令の数をロードすることができる。この形で、完了した命令の数をカウントすることができる。

【0031】命令が取り消される場合、完了テーブル650内のその項目は、再割振りされ、完了しなかった命

令のイベントの発生の関連回数は、失われる。割振りタグ634には、リセット信号621など、適当な仮カウンタの値をゼロにする、その仮カウンタへのリセット信号を送るためにデコーダ622を制御するのに使用される、完了テーブル650へのインデックスが格納される。

【0032】PMC1に類似の形で、フィールド676は、制御信号681を生成し、その結果、PMC2 684が仮カウンタから仮回数を受け入れるようにするのに使用される。仮カウンタに格納されたイベントの仮回数は、加算器682によってPMC2 684の元の値に加算され、その後、PMC2 684に格納される。マルチプレクサ680が、制御信号681によって適当に制御される場合、イベント688によって、適当な回数値（図示せず）が加算器682に送られ、加算器682はPMCを更新する。

【0033】上で説明した例では、PMC1によって、完了する命令に関する指定されたイベントの発生がカウントされる。本発明によって提供される形で、PMC2は、すべての命令に関するイベントをカウントすることができ、これには、完了する命令の実行に関するイベントと、投機的に実行され、完了しない命令の実行に関するイベントが含まれる。この形で、本発明は、実際に完了する命令によって引き起こされるイベントだけをカウントする方法を提供する。PMCによってカウントできるイベントに、仮カウンタならびに仮カウンタに経路指定されるイベントが含まれる場合、一部のPMCですべての命令に関する総イベント回数を累算し、他のPMCで、完了する命令に関する非投機的イベント回数だけを累算することができる。総回数と非投機的回数との差が、取り消された投機的に実行された命令によって引き起こされたイベント発生の数である。

【0034】図7を参照すると、流れ図に、本発明の実施例による投機的プロセッサの性能モニタ内で、完了する命令によって引き起こされた指定されたイベントのイベント発生回数（非投機的イベント回数）を累算するための処理が示されている。この処理は、命令の取出と復号によって開始される（ステップ702）。取り出され、復号された命令のために完了テーブルの項目を割り振った（ステップ704）後に、完了テーブル項目に関連する仮カウンタをリセットする（ステップ706）。特定の命令が実行されている間に、使用可能にされたイベントが発生したかどうかに関する判定を行う（ステップ708）。そうである場合には、完了テーブル項目に関連する仮カウンタをインクリメントし（ステップ710）、この処理は、ループ・バックしてプロセッサの実行の監視を継続する。

【0035】使用可能にされたイベントが発生しなかった場合には、命令が実行を完了したかどうかに関する判定を行う（ステップ712）。そうである場合には、仮

カウンタを適当なPMCに加算し、この処理は、ループ・バックして、別の命令を取り出し、復号する。命令が完了しなかった場合には、命令が取り消されたかどうかに関する判定を行う（ステップ716）。そうでない場合には、この処理は、ループ・バックして、プロセッサの実行の監視を継続する。命令が取り消された場合には、プロセッサが停止したかどうかに関する判定を行う（ステップ718）。そうでない場合には、この処理は、ループ・バックして、別の命令を取り出し、復号する。プロセッサが停止している場合には、この処理は終了する。

【0036】その代わりに、この処理は、処理中の複数の点で取出と復号にループ・バックすることができる。重要な点は、この流れ図に、プロセッサの性能監視機能が使用可能にされている間に論理的に実行される、基本的に終了しないループが示されていることである。

【0037】図8を参照すると、流れ図に、投機的プロセッサの性能モニタを使用して、投機的に実行された命令によって引き起こされた指定されたイベントのイベント発生回数を計算するための処理が示されている。上で述べたように、性能モニタ内の特殊なレジスタは、プロセッサ内のハードウェアの性能を分析するソフトウェア・パッケージなどのソフトウェアからアクセス可能である。

【0038】この処理は、ソフトウェアが、完了した命令に関連するイベントの回数の責任を負うPMCを読み取る時に開始される（ステップ802）。その後、このソフトウェアは、すべての命令に関連するイベントの回数の責任を負うPMCを読み取る（ステップ804）。その後、PMCから読み取った値の間の差を計算し（ステップ806）、投機的に実行された命令に関するイベントの回数について出力値を生成する（ステップ808）。

【0039】本発明がもたらす長所は、上に示した本発明の詳細な説明に鑑みて明白である。いくつかの条件の下では、完了はしないが部分的な実行ステージ中にやはりプロセッサ資源を消費する命令の投機的実行によって、投機的プロセッサの性能が大幅に低下する可能性がある。本発明は、投機的に実行される命令の性能を監視するための方法およびシステムを提供し、その結果、システム設計者が、命令の投機的実行によってプロセッサの性能が大幅に低下しないようにすることができるようにする。

【0040】たとえば、投機的命令は、メモリ帯域幅などのプロセッサ内の資源を消費する。メモリ・アクセス動作の性能は、特に第1レベル・キャッシュでミスが発生する時に、重要な検討事項になる。というのは、コンピュータ・システムが、システムのコストに対する敏感さに起因して、階層メモリ・サブシステムを使用することを強制されるからである。一般に、メモリ階層は、ア

クセスされるすべてのメモリ位置で、汎用レジスタほど高速の理想的な性能を達成できない。階層メモリ・システムの鍵となる概念は、メモリ・アクセスが、通常は第1レベル・キャッシュ・アクセスによって即座に満足されることである。予想されるとおり、頻繁ではないがコストの高い第1レベル・キャッシュ・ミスが存在する。第1レベル・キャッシュ・ミスが存在する状況でプロセッサが効率的に実行できることが望ましい。したがって、設計目標の1つが、できる限り効率的にメモリ階層を利用することになる。

【0041】階層メモリ・システムを効率的に使用方法の1つが、アウトオブオーダー実行を実施することである。これによって、第1レベル・キャッシュ・ミスをこうむる命令の付近の命令に、メモリ・アクセス命令がこれらのミスによって遅延される期間中に実行する機会が与えられる。完了ユニットの主な仕事の1つが、アーキテクチャ的に有効な形で、アウトオブオーダー実行される命令を並べなおす(reorder)ことである。命令のアウトオブオーダー実行に対する主な障害の1つが、アーキテクチャ的依存性を処理するための命令の直列化である。スーパースカラ・プロセッサは、命令レベルの並列性を有することができるので、単に第1レベル・キャッシュ・ミスが存在することを知らただけでは不十分である。というのは、第1レベル・キャッシュ・ミスが、特に命令の完了を遅延させる時に、他の命令の完了の遅延を引き起こさない可能性があるからである。アウトオブオーダー実行のまさにこの性質が、これを確認することを困難にしている。

【0042】本発明がもたらす長所の1例として、アウトオブオーダー実行の完了に対する遅延の影響を、完了する投機的に実行された命令の第1レベル・キャッシュ・ミスに起因して失われるサイクルについて測定することができる。そのようなストールを生じる特定の問題が識別され、その相対的な重要性が確認された後に、そのデータを使用して、コンパイラの変更、ソフトウェアの変更またはハードウェア設計の変更もしくはこれらの組合せを行うことができる。一般に、これは、分岐ユニット動作の性能を監視する方法の1つである。分岐予測は、分岐ユニット動作の大きな部分である。完了しない命令による資源の使用に関するイベントの回数を追跡することによって、予測論理の有効性を評価することができる。

【0043】完全に機能するデータ処理システムに関連して本発明を説明してきたが、本発明の処理を、命令のコンピュータ可読媒体の形およびさまざまな形で配布ことができ、本発明が、配布の実行に実際に使用される信号担持媒体の具体的な種類に無関係に同等に適用されることを、当業者が諒解するであろうことに留意することが重要である。コンピュータ可読媒体の例には、フロッピー・ディスク、ハード・ディスク装置、RAMおよ

びCD-ROMなどの記録可能型媒体と、デジタル通信リンクおよびアナログ通信リンクなどの伝送型媒体が含まれる。

【0044】本発明の説明は、例示と説明の目的のために提示されたものであり、網羅的であることを意図したものではなく、開示された形態に本発明を制限することを意図したものでもない。多数の修正形態および変形形態が、当業者には明白であろう。実施例は、本発明の原理と実用的な応用例を最もよく説明し、意図された特定の使用に適するさまざまな修正を加えたさまざまな実施形態に関して当業者が本発明を理解するために選択され、記述されたものである。

【0045】まとめとして、本発明の構成に関して以下の事項を開示する。

【0046】(1) 投機的プロセッサと複数のカウンタとを含むデータ処理システム内でイベントを監視する方法であって、前記複数のカウンタの中の各カウンタが、前記データ処理システム内の指定されたイベントの発生をカウントし、監視するイベントを指定し、前記投機的プロセッサによる命令の実行中に前記指定されたイベントについて監視し、前記投機的プロセッサによって実行されるすべての命令について、前記指定されたイベントの発生回数値を生成し、前記投機的プロセッサによって完全に実行される命令について、前記指定されたイベントの発生回数を生成する方法。

(2) すべての命令に関する前記指定されたイベントの発生の前記回数と、すべての完了した命令に関する前記指定されたイベントの発生の前記回数との間の差を、前記投機的プロセッサによって投機的に実行された命令の前記指定されたイベントの発生回数として計算するステップをさらに含む、上記(1)に記載の方法。

(3) 前記投機的プロセッサによって実行される命令のそれぞれについて複数の指定されたイベントを監視するステップをさらに含む、上記(1)に記載の方法。

(4) 投機的プロセッサと複数のカウンタとを含むデータ処理システム内でイベントを監視する方法であって、前記複数のカウンタの中の各カウンタが、前記データ処理システム内の指定されたイベントの発生をカウントし、仮カウンタを特定の命令に関連付け、第1グローバル・イベント・カウンタをすべての命令に関連付け、第2グローバル・イベント・カウンタを完了した命令に関連付け、監視するイベントを指定し、前記投機的プロセッサによる命令の実行中に前記指定されたイベントについて監視し、前記特定の命令の実行中の前記イベントの発生の検出にตอบสนองして、前記仮カウンタをインクリメントし、命令のいずれかの実行中の前記イベントの発生の検出にตอบสนองして、前記第1グローバル・イベント・カウンタをインクリメントし、前記特定の命令の完了の検出にตอบสนองして、前記仮カウンタからのイベント回数を前記第2グローバル・イベント・カウンタに加算する方法。

(5) 前記第1グローバル・イベント・カウンタからのイベント回数と、前記第2グローバル・イベント・カウンタからのイベント回数との間の差を、前記投機的プロセッサによって投機的に実行された命令に関する前記指定されたイベントの発生回数として計算するステップをさらに含む、上記(4)に記載の方法。

(6) 投機的プロセッサと複数のカウンタとを含むデータ処理システム内でイベントの回数を計算する方法であって、前記複数のカウンタの中の各カウンタが、前記データ処理システム内の指定されたイベントの発生をカウントし、前記投機的プロセッサによって実行されるすべての命令に関する指定されたイベントの発生回数を第1カウンタから読み取り、前記投機的プロセッサによって完全に実行される命令に関する指定されたイベントの発生回数を第2カウンタから読み取り、すべての命令に関する前記指定されたイベントの発生の前記回数と、すべての完了した命令に関する前記指定されたイベントの発生の前記回数との間の差を、前記投機的プロセッサによって投機的に実行された命令に関する前記指定されたイベントの発生回数として計算する方法。

(7) 投機的プロセッサと複数のカウンタとを含むデータ処理システム内でイベントを監視する方法であって、前記複数のカウンタの中の各カウンタが、前記データ処理システム内の指定されたイベントの発生をカウントし、監視するイベントを指定し、前記指定されたイベントを監視し、特定の指定されたイベントの発生の検出にตอบสนองして、第1カウンタおよび第2カウンタをインクリメントし、命令の完了の検出にตอบสนองして、前記第2カウンタを第3カウンタに加算する方法。

(8) 前記第1カウンタが、前記投機的プロセッサによって実行されるすべての命令に関する特定の指定されたイベントの発生をカウントする、上記(7)に記載の方法。

(9) 前記第2カウンタが、特定の命令に関する特定の指定されたイベントの発生をカウントする、上記(7)に記載の方法。

(10) 前記第3カウンタが、前記投機的プロセッサによって完全に実行される命令に関する特定の指定されたイベントの発生をカウントする、上記(7)に記載の方法。

(11) 投機的に実行された命令に関する特定の指定されたイベントの発生回数を生成するために、前記第1カウンタと前記第3カウンタとの間の差を計算するステップをさらに含む、上記(7)に記載の方法。

(12) 投機的プロセッサと複数のカウンタとを含むデータ処理システム内でイベントを監視するための装置であって、前記複数のカウンタの中の各カウンタが、前記データ処理システム内の指定されたイベントの発生をカウントし、監視するイベントを指定するための手段と、前記投機的プロセッサによる命令の実行中に前記指定さ

れたイベントについて監視するための手段と、前記投機的プロセッサによって実行されるすべての命令に関する前記指定されたイベントの発生回数を生成するための第1手段と、前記投機的プロセッサによって完全に実行される命令に関する前記指定されたイベントの発生回数を生成するための第2手段とを含む装置。

(13) すべての命令に関する前記指定されたイベントの発生の前記回数と、すべての完了した命令に関する前記指定されたイベントの発生の前記回数との間の差を、前記投機的プロセッサによって投機的に実行された命令に関する前記指定されたイベントの発生回数として計算するための手段をさらに含む、上記(12)に記載の装置。

(14) 前記投機的プロセッサによって実行される命令のそれぞれについて複数の指定されたイベントを監視するための手段をさらに含む、上記(12)に記載の装置。

(15) 複数のカウンタを含む投機的プロセッサ内のイベントの回数を計算するためのシステムであって、前記複数のカウンタの中の各カウンタが、前記投機的プロセッサ内の指定されたイベントの発生をカウントし、前記投機的プロセッサによって実行されるすべての命令に関する指定されたイベントの発生回数を第1カウンタから読み取るための第1手段と、前記投機的プロセッサによって完全に実行される命令に関する前記指定されたイベントの発生回数を第2カウンタから読み取るための第2手段と、すべての命令に関する前記指定されたイベントの発生の前記回数と、すべての完了した命令に関する前記指定されたイベントの発生の前記回数との間の差を、前記投機的プロセッサによって投機的に実行された命令に関する前記指定されたイベントの発生回数として計算するための手段とを含むシステム。

(16) 投機的プロセッサと複数のカウンタとを含むデータ処理システム内でイベントを監視するための、コンピュータ可読媒体上のコンピュータ・プログラム製品であって、前記複数のカウンタの中の各カウンタが、前記データ処理システム内の指定されたイベントの発生をカウントし、監視するイベントを指定するための第1命令と、前記投機的プロセッサによる命令の実行中に前記指定されたイベントについて監視するための第2命令と、前記投機的プロセッサによって実行されるすべての命令に関する前記指定されたイベントの発生回数を生成するための第3命令と、前記投機的プロセッサによって完全に実行される命令に関する前記指定されたイベントの発生回数を生成するための第4命令とを含むコンピュータ・プログラム製品。

(17) 複数のカウンタを含む投機的プロセッサ内でイベントの回数を計算するための、コンピュータ可読媒体上のコンピュータ・プログラム製品であって、前記複数のカウンタの中の各カウンタが、前記投機的プロセッサ

内の指定されたイベントの発生をカウントし、前記投機的プロセッサによって実行されるすべての命令に関する指定されたイベントの発生回数を第1カウンタから読み取るための第1命令と、前記投機的プロセッサによって完全に実行される命令に関する前記指定されたイベントの発生回数を第2カウンタから読み取るための第2命令と、すべての命令に関する前記指定されたイベントの発生の前記回数と、すべての完了した命令に関する前記指定されたイベントの発生の前記回数との間の差を、前記投機的プロセッサによって投機的に実行された命令に関する前記指定されたイベントの発生回数として計算するための第3命令とを含むコンピュータ・プログラム製品。

【図面の簡単な説明】

【図1】本発明の好ましい実施例による、本発明を実施することのできるデータ処理システムを表す絵図である。

【図2】本発明の好ましい実施例による、情報を処理するためのデータ処理システムの選択された内部機能ユニットを示すブロック図である。

【図3】2つの性能モニタ・カウンタの動作を制御するのに適したモニタ・モード制御レジスタの1つの構成の例を表す図である。

【図4】アウトオブオーダー投機的実行プロセッサ内の命令パイプラインのステージの詳細を示すブロック図である。

【図5】投機的プロセッサ内で使用することのできる命令完了テーブルを示す図である。

【図6】本発明の方法による、完了した命令に関するイベントから分離して、投機的に実行された命令に関するイベントをカウントするためのプロセッサを示す図である。

る。

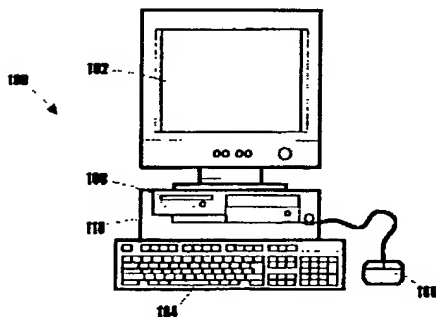
【図7】本発明の実施例による投機的プロセッサの性能モニタ内で、完了した命令によって引き起こされた指定されたイベントのイベント発生回数（非投機的イベント回数）を累算するための処理を示す流れ図である。

【図8】投機的プロセッサの性能モニタを使用して、投機的に実行された命令によって引き起こされた指定されたイベントのイベント発生回数を計算するための処理を示す流れ図である。

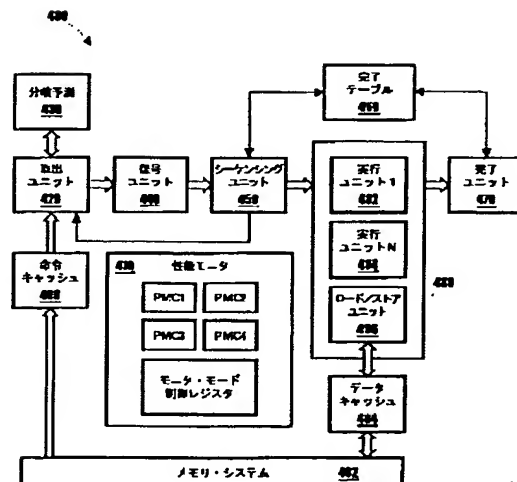
【符号の説明】

- 610 マルチプレクサ
- 612 イベント信号
- 614 イベント・タグ
- 616 デコーダ
- 618 インクリメンタ
- 622 デコーダ
- 630 マルチプレクサ
- 632 完了タグ
- 634 割振りタグ
- 650 完了テーブル
- 652 タグ値
- 654 仮カウンタ
- 660 マルチプレクサ
- 662 加算器
- 664 PMC 1
- 668 イベント
- 670 MMCR
- 680 マルチプレクサ
- 682 加算器
- 684 PMC 2
- 688 イベント

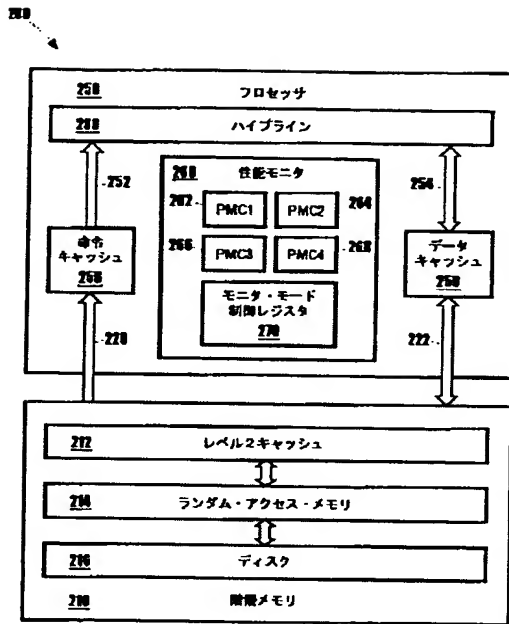
【図1】



【図4】



【図2】



【図5】

テーブル 項目	命令の種類	ソース	デスティネーション
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			

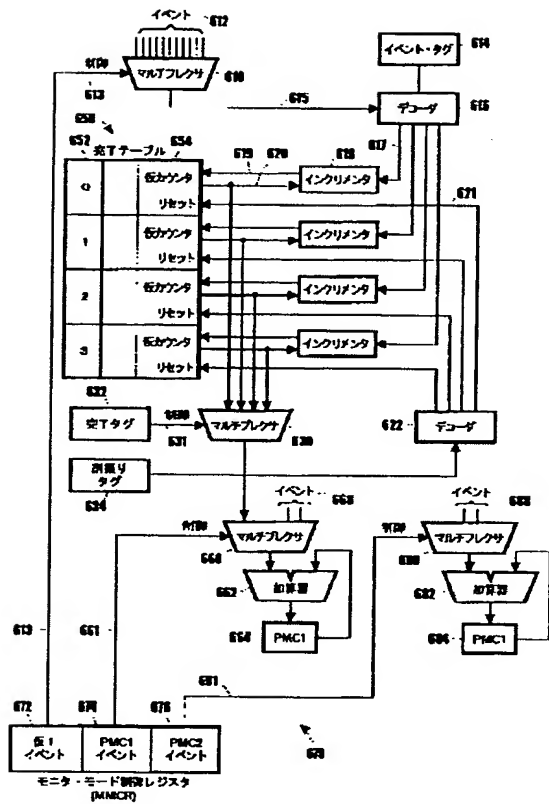
図5は、命令テーブルの構造を示す図である。左側に「完了」および「割り込み」のラベルがあり、右側のテーブルの各行に対応している。

【図3】

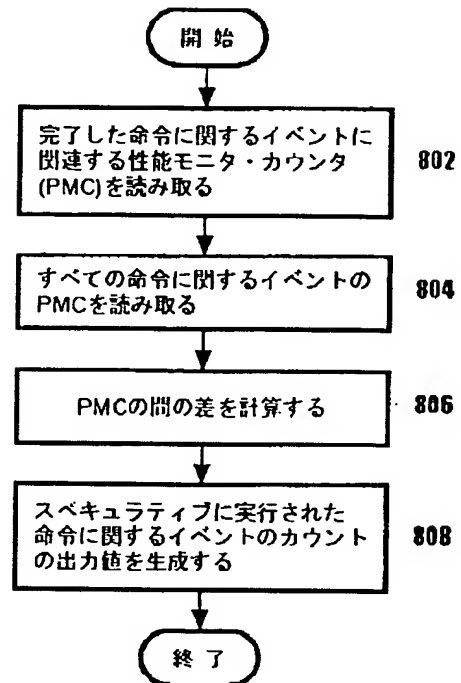
ビット0~4 カウント 使用可能	ビット5 割込み 使用可能	ビット 6~15	ビット16 PMC2 割込み制御	ビット17 PMC2 割込み制御	ビット18 PMC2 カウント制御	ビット19~25 PMC1 イベント選択	ビット26~31 PMC2 イベント選択
------------------------	---------------------	-------------	------------------------	------------------------	-------------------------	----------------------------	----------------------------

モニタ・モード制御レジスタ
(MMCR)

【例6】



【図8】



【 図 7 】

